

METHOD, SYSTEM, AND PROGRAM FOR ERROR  
HANDLING IN A DUAL ADAPTOR SYSTEM

RELATED APPLICATIONS

5 [0001] This application is related to the copending and commonly assigned United States patent application entitled "Method, System, and Program for Error Handling in a Dual Adaptor System Where One Adaptor Is a Master", having attorney docket no. TUC920010104US1, which patent application was filed on the same date herewith and is incorporated herein by reference in its entirety.

10

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0002] The present invention relates to a system, method, and program for error handling in a dual adaptor system.

15

2. Description of the Related Art

[0003] In a storage loop architecture, such as the Serial Storage Architecture (SSA), a plurality of disks are interconnected to one or more adaptors so that either of the adaptors can access the one or more loops of interconnected disks. An adaptor may include two or more ports to allow connection to one or more loops. For each loop on which the adaptor communicates, one adaptor port connects to a first disk in the loop and the other port connects to another disk in the loop. Additional adaptors may be added to the loop, such that one port on each other adaptor connects to one disk and another port connects to another disk so that the additional adaptors are placed within the loop. Additional details of the SSA architecture and different possible loop topologies are described in the International Business Machines Corporation (IBM) publication "Understanding SSA Subsystems in Your Environment", IBM document no. SG24-5750-00 (April, 2000), which publication is incorporated herein by reference in its entirety.

[0004] One or more computer systems, such as storage subsystems, host system, etc., may include the adaptors connecting to the loop. Adaptors that share a loop must intercommunicate to coordinate accesses to disks in the shared loop. High end storage systems, such as the IBM Enterprise Storage Server (ESS), can detect errors in the ability 5 of an adaptor in another system to communicate with the local operating system even though such detected adaptor is still capable of communicating on the network. In such instances, the system detecting the problem will delay I/O processing for a timeout period that corresponds to the time required for the other system including the adaptor to initiate an error recovery procedure. This timeout period must take into account all different 10 timeout periods and error recovery procedures that could occur within the detected system unable to communicate with the adaptor. In many cases the timeout period can extend for several minutes.

[0005] In storage systems requiring high availability, such as storage systems for critical uses, any delays in I/O processing are generally unacceptable. Thus, extensive 15 delays in I/O processing, such as a delay resulting from the lengthy timeout period for the error recovery process at the detected system, would be unacceptable in a high availability system.

[0006] For these reasons there is a need in the art to provide improved error handling that reduces timeout delays in systems where two adaptors are capable of accessing the 20 storage devices.

#### SUMMARY OF THE PREFERRED EMBODIMENTS

[0007] Provided is a method, system, and program for processing Input/Output (I/O) requests to a storage network including at least one storage device and at least two 25 adaptors, wherein each adaptor is capable of communicating I/O requests to the at least one storage device. An error is detected in a system including a first adaptor, wherein the first adaptor is capable of communicating on the network after the error is detected. In response to detecting the error, a monitoring state is initiated to monitor I/O requests transmitted through a second adaptor. In response to receiving an I/O request, an I/O

delay timer is started that is less than a system timeout period. After the system timeout period the error recovery process in the system including the first adaptor would complete. A reset request is sent to the first adaptor in response to detecting an expiration of one started I/O delay timer.

- 5 [0008] In certain implementations, the detected error indicates that the first adaptor is unable to communicate with the system including the first adaptor.
- [0009] Still further, I/O requests may continue to be processed through the second adaptor until the reset request is sent.
- [0010] In still further implementations, a monitoring timer equivalent to the system 10 timeout period is started after detecting the error at the first adaptor. The monitoring state and any pending I/O delay timers are terminated after the monitoring timer expires.
- [0011] Yet further, the steps of initiating a monitoring state, starting the I/O delay timer and sending the reset request may be performed by a device driver executing in an operating system.
- 15 [0012] The described implementations provide improved error handling techniques when detecting a problem with the ability of another adaptor to communicate to the system in which the adaptor is housed. The described error handling techniques reduce delays in I/O processing when detecting errors in the ability of other adaptors to communicate to the system in which they are housed.

20

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 illustrates a computing environment in which aspects of the invention are 25 implemented; and

FIG. 2 illustrates an alternative computing environment in which aspects of the invention are also implemented; and

FIG. 3 illustrates logic implemented when detecting a problem in the system housing another adaptor in accordance with implementations of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0014] In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several embodiments of the present invention. It is understood that other embodiments may be utilized and structural and 5 operational changes may be made without departing from the scope of the present invention.

[0015] FIG. 1 illustrates a computing environment in which aspects of the invention are implemented. Computer systems 2a and 2b include at least one adaptor 4a and 4b to enable communication with a loop 6 of interconnected storage devices 8a, 8b...8j, such as 10 hard disk drives, tape drives, optical disk drives, etc. Each adaptor 4a and 4b includes a plurality of ports 10a, 12a, 14a, 16a and 10b, 12b, 14b, and 16b, respectively. In FIG. 1, ports 10a, 10b, 12a, and 12b are shown as connected to the loop 6. The other ports 14a, 14b, 16a, and 16b may connect to additional loops. Further, FIG. 1 provides an example 15 loop configuration. Those skilled in the art will appreciate that there are numerous loop configurations that may be implemented between one or more adaptors.

[0016] The systems 2a and 2b may comprise any computing device known in the art, such as a host system, storage subsystem, workstation, server, etc. Each system 2a and 2b includes an operating system 20a and 20b, respectively, and device drivers 22a and 22b that enable communication between the operating systems 20a, 20b and the adaptors 20 4a, 4b. Each of the storage devices 8a, 8b...8j would also include ports (not shown) to connect to the loop 6. In certain implementations, the loop 6 may be implemented using the Storage Serial Architecture (SSA) or any other protocol enabling multiple hosts to communicate with multiple storage device, such as Fibre Channel, Ethernet, Gigabyte Ethernet, Infiniband, etc. The loop 6 may be part of a Storage Area Network (SAN), 25 Wide Area Network (WAN), Local Area Network (LAN), the Internet, an Intranet, etc.

[0017] FIG. 2 illustrates an alternative implementation where the adaptors 54a and 54b are housed in different Input/Output bays 52a and 52b within the same system 50. Each I/O bay 52a and 52b provides a separate power boundary. Thus, adaptors 54a and 54b are on separate power boundaries. Each adaptor 54a and 54b includes ports 60a, 62a,

64a, 66a and 60b, 62b, 64b, and 66b, respectively, for communication with one or more loops (not shown) according to any loop topology known in the art. In the implementation of FIG. 2, the system 50 includes one operating system 70 and separate device drivers 72a and 72b to manage the operations of adaptors 54a and 54b.

5 Alternatively, only one instance of the device driver may execute to manage the operations of both the adaptors 54a and 54b. Each I/O bay 52a and 52b provides different communication channels and power sources for the adaptors 54a and 54b, respectively.

[0018] FIG. 3 illustrates logic implemented in the device drivers 22a and 22b (FIG. 1),  
10 as well as device drivers 72a and 72b (FIG. 2) to handle a situation where an error is detected in the system housing the other target adaptor 4b. For instance, the target adaptor 4b may be unable to communicate with the operating system 20b due to a problem in the bus interfacing between the adaptor 4b and the system 2b in which the operating system 20b is executing. For instance, the adaptor 4b may be implemented on  
15 a Peripheral Component Interconnect (PCI) card and the PCI bus between the system 2b and adaptor 4b may be malfunctioning. In such cases, the target adaptor 4b itself is operational, but does not fully function because of another problem in the system 2b housing the target adaptor 4b.

[0019] The error handling process of FIG. 3 begins at block 100 where the device driver in one adaptor, e.g., device driver 22a in adaptor 4a, detects an error within the system 2b including the target adaptor 4b, even though no error was detected at the target adaptor 4b. In such case, the source adaptor 4a would be able to continue processing I/Os against the storage devices 8a, 8b...8j because the target adaptor 4b is available on the loop 6. In response to detecting the error, the device driver 22a begins two  
25 concurrent (or non-concurrent) processes at blocks 102 and 106. At block 102, the device driver 22a starts a window timeout timer 102. In certain implementations, the duration of the window timeout timer coincides with a time during which the system 2b including the target adaptor 4b is guaranteed to have completed an error recovery procedure. This window timeout timer includes the time that would pass before the other

system 2b would initiate the error recovery procedure after detecting a problem. The error recovery procedure initiated by the system 2b would likely eliminate the detected error. Upon the system 2b beginning the error recovery procedure by resetting or undergoing a power cycle, the target adaptor 4b would go offline and not participate in

5 the loop 6. In certain implementations, each adaptor 4a, 4b on the loop is aware of all active adaptors. Once an adaptor becomes unavailable or unable to communicate on the loop 6, then the adaptors remaining on the loop are notified of such state change and remove the unavailable adaptor from loop registries (not shown) each adaptor maintains that indicates available adaptors on the loop. The capability of one system to detect

10 errors in other systems, such as problems a remote system may have in communicating with internal adaptors is known in the art of high availability storage subsystems, such as the IBM Enterprise Storage Servers (ESS).\*\* Thus, if the device driver 22a, 22b takes no action for the window timeout period, the device driver 22a, 22b can be assured that the other system 2b will have reset and the communication problem with the target

15 adaptor 4b will have been repaired or taken off-line. Thus, after an alert is received (at block 104) that the windows timeout timer expired, the monitoring ends and any processes or timers started in response to receiving an I/O request at block 104 are terminated (at block 105) because the communication problem in the system 2b including the target adaptor 4b is in the process of being repaired. During the system 2b error

20 recovery procedure, the target adaptor 4b would go off-line and not affect loop operations.

[0020] At block 106, upon receiving an Input/Output (I/O) request, the device driver 22a starts (at block 108) an I/O delay timer for the received I/O request and forwards the I/O request to the adaptor 4a to execute against the target storage device 8a, 8b....8j. In

25 certain implementations, the I/O delay timer reflects a maximum tolerable I/O delay dependent on the environment in which the storage system is deployed. Thus, the I/O delay timer in high availability systems is relatively shorter than systems requiring less availability. For instance, in high availability systems, the windows timeout period may run for several minutes to ensure the initiation of the error recovery process at the remote

system 2b and the I/O delay period may be only a couple of seconds. Upon receiving (at block 110) notification that the monitored I/O request completed, the device driver 22a would stop (at block 112) the I/O delay timer that was started for the completed I/O request and control returns to block 106 to await any further received I/O requests or one of the other events at blocks 104 and 114. Upon receiving (at block 114) notification that one I/O delay timer expired, the device driver 22a issues (at block 116) a command to the local adaptor 4a to get the identifier (ID) of the target adaptor 4b. As mentioned, in certain implementations, the local adaptor 4a could access the target adaptor 4b from a loop registry. If there is no ID for the target adaptor 4b in the loop registry, then that target adaptor 4b is not available on the loop. However, the unavailable target adaptor may still affect I/O operations of the source adaptor 4a, such as by preventing the loop 6 from completing a configuration. The target adaptor 4b may be only temporarily unavailable. If (at block 118) the get ID operation succeeded, i.e., the ID of the target adaptor 4b is included in the loop registry, then the device driver 22a issues (at block 120) a reset request to the received ID of the target adaptor 4b to cause a reset of the target adaptor 4b. Otherwise, if (at block 118) the get ID request did not succeed, then the device driver 22a returns to block 116 to issue additional get ID requests until the ID is received or the window timeout timer expires at block 104. The device driver 22a may be unable to obtain the ID if the system 2b including the target adaptor 4b, or the I/O bays 52a and 52b in the case of FIG. 2, is in the process of being reset.

**[0021]** In alternative implementations, the source adaptor 4a may query the target adaptor 4b over the loop 6 for the adaptor 4b ID. With the logic of FIG. 3, if the target adaptor 4b becomes available after one or more get ID requests fail at block 118 and before the windows timeout timer expires at block 104, then the device driver 22a would send the reset command to the adaptor 4b to cause the adaptor 4b to reset. If, during the reset operation, the target adaptor 4b is unable to establish communication with the system 2b, then the target adaptor 4b would not come back online and would not be available on the loop 6 until after the system 2b resets or power cycles and corrects the communication problem with the target adaptor 4b. If the target adaptor 4b becomes

unavailable as a result of the reset, then the source adaptor 4a and any other adaptors on the loop 6 can continue operating on the loop 6 without concern that the problems at the off-line target adaptor 4b will cause problems.

[0022] In the described implementations, the device drivers 22a and 22b communicate 5 an adaptor reset command. In alternative implementations, a system reset or power cycle command may be communicated. In implementations such as FIG. 2 where the two adaptors are housed in the same system, then a problem may occur at the I/O bay 52a and 52b level including the adaptor, such that the adaptor, e.g., adaptor 54b cannot communicate with the system even though the adaptor 54b can communicate on the loop.

10 In such case, the device driver 72a detecting the problem in the I/O bay in the same system 50 may perform the reset with respect to the specific I/O bay 52b where the error is detected to avoid resetting or rebooting the entire system 50.

[0023] With the logic of FIG. 3, the device drivers 22a, 22b upon detecting errors at another peer adaptor will begin monitoring I/O requests to determine if any unacceptable 15 I/O delays have occurred, i.e., I/O delays that extend the duration of the I/O delay timer. If no such unacceptable I/O delays occur, then the device drivers 22a, 22b would continue performing I/Os and the system 2a, 2b housing the target adaptor would eventually reset without affecting I/O processing at the other adaptor 4a, 4b. However, if an I/O request is delayed for the deemed unacceptable period, i.e., the I/O delay period, 20 then a reset request will be sent to the adaptor 4b including the target adaptor 4b under the assumption that the problems in the system 2b including the target adaptor 4b are affecting the target adaptor's 4b ability to communicate on the loop 6. In cases where the device driver 22a, 22b cannot communicate with the target adaptor, i.e., cannot obtain the ID of the target at blocks 116-120, then the device driver 22a will continue to attempt to 25 obtain the target adaptor 4b ID by resending the get ID request until either the identifier is received so that a reset request can be transmitted at block 120 or the window timeout timer expires at block 104 indicating that the system 4b housing the detected adaptor 4b with the communication problem has had sufficient time to initiate and complete an error recovery procedure.

[0024] The logic of FIG. 3 thus provides a greater degree of intelligent monitoring in cases where there is a problem in the system including one adaptor that will eventually result in a reset after a period of time that is longer than an acceptable I/O delay time given the availability requirements of the system. With the described implementations, 5 unacceptable I/O delays are not experienced while waiting for the system or I/O bay including the other adaptor to reset. The described implementations are particularly suited for situations where the detected problem between the target adaptor 4b and system 2b including the adaptor is not likely to prevent I/O requests at other adaptors from completing. The logic of FIG. 3 would allow one adaptor to continue processing 10 I/O requests and wait for the detected adaptor to internally reset without having to interrupt I/O processing.

Additional Implementation Details

[0025] The preferred embodiments may be implemented as a method, apparatus or 15 article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term “article of manufacture” as used herein refers to code or logic implemented in hardware logic (e.g., an integrated circuit chip, Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium (e.g., magnetic storage 20 medium (e.g., hard disk drives, floppy disks,, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.). Code in the computer readable medium is accessed and executed by a processor. The code in which preferred embodiments are implemented may further be accessible through a 25 transmission media or from a file server over a network. In such cases, the article of manufacture in which the code is implemented may comprise a transmission media, such as a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Of course, those skilled in the art will recognize that many modifications may be made to this configuration without departing

from the scope of the present invention, and that the article of manufacture may comprise any information bearing medium known in the art.

[0026] In the described implementations, the monitoring logic was implemented in the adaptor device driver, which is an operating system program. Alternatively, the 5 monitoring logic of FIG. 3 may be implemented within the adaptor hardware.

[0027] In the described implementations, the detected error condition which triggered the logic of FIG. 3 indicated a problem in the system including the adaptor, where the adaptor itself is functioning properly. However, the logic of FIG. 3 may be initiated in response to other errors, such as internal adaptor hardware errors, line fault errors, ACK 10 time out error in not receiving the required acknowledgment from the target adaptor, loss of synchronization error, code violation error in decoding a character that is not data or is a special character, protocol error, sequence error, frame reject error, etc. Further details of the errors that may trigger the error recovery logic of FIG. 3 are described in the publication "Serial Storage Architecture - Transport Layer 2 (SSA-TL2)", Revision 5b, 15 Working Draft T10.1/1147D (ANSI, April 4, 1997), which publication is incorporated herein by reference in its entirety.

[0028] In the described implementations, the reset sent by the device drivers 22a and 22b is a reset to the adaptor and not other components in the system housing the adaptor. In alternative implementations, the reset may cause a reset or power cycle of the entire 20 system housing the adaptor. In implementations where the system including the adaptor where the error is detected includes multiple power boundaries, i.e., multiple I/O bays, then the reset may cause a power cycle to the specific power boundary, i.e., I/O bay, including the target adaptor 4b where the problem is detected. Such a power boundary specific reset would not affect other system components, such as other adaptors, on other 25 power boundaries, such as other I/O bays.

[0029] In the described implementations, the storage devices and adaptors are all connected on a loop topology, using a loop protocol such as SSA, the Fibre Channel arbitrated loop protocol, etc. In alternative implementations, the two adaptors may be connected to the storage devices in a non-loop topology. In the topology shown in FIG.

1, each adaptor may directly communicate with any of the storage devices 8a, 8b...8j. In additional implementations, one adaptor may have to communicate through another adaptor to reach a target storage device.

[0030] In certain implementations, the disk drives 8a, 8b...8j comprised magnetic hard disk drives. In alternative implementations, the storage devices 8a, 8b...8j may comprise any storage device known in the art, such as optical disks, tapes, etc.

[0031] The preferred logic of FIG. 3 describes specific operations occurring in a particular order. In alternative implementations, certain of the logic operations may be performed in a different order, modified or removed. Moreover, steps may be added to the 10 above described logic and still conform to the described implementations. Further, operations described herein may occur sequentially or certain operations may be processed in parallel.

[0032] The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be 15 exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto. The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many 20 embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

---

\*\*IBM and Enterprise Storage Server are trademarks of International Business Machines Corporation.